UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/746,796	12/22/2000	Ariel Cohen	00-162 1496.0047	9162

22501 7590 03/30/2010

CHRISTOPHER P MAIORANA, PC
LSI Corporation
24840 HARPER
SUITE 100
ST CLAIR SHORES, MI 48080

EXAMINER	
GEIB, BENJAMIN P	
ART UNIT	PAPER NUMBER
2181	

MAIL DATE	DELIVERY MODE
03/30/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ARIEL COHEN, RONEN PERETS and
BORIS ZEMLYAK

Appeal 2008-005686
Application 09/746,796
Technology Center 2100

Decided: March 30, 2010

Before JOHN A. JEFFERY, HOWARD B. BLANKENSHIP, and
DEBRA K. STEPHENS, *Administrative Patent Judges*.

STEPHENS, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) (2002) from a final rejection of claims 1-22 (App. Br. 3). We have jurisdiction under 35 U.S.C. § 6(b) (2008).

We AFFIRM-IN-PART.

Introduction

According to Appellants, the invention is a system and method for "a microcode based hardware translator to support a multitude of processors" (Spec. 1, ll. 15-16). A circuit translates instruction codes of a first instruction set into sequences of a second instruction set's instruction codes (Abst.). The sequences of instruction codes emulate functionality of the first instruction set's instruction codes (*id.*).

STATEMENT OF THE CASE

Exemplary Claim

Claim 1 is an exemplary claim and is reproduced below:

1. An apparatus comprising:

> a circuit configured to translate instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set.

Prior Art

Martin	4,439,828	Mar. 27, 1984
Hilgendorf	5,925,124	Jul. 20, 1999
Gee	6,374,286 B1	Apr. 16, 2002

Rejections

Claims 1-6, 8, 9, 12-14, and 17-19 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Hilgendorf.

Claims 10, 11, and 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf and Martin.

Claims 7 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf.

Claims 16 and 21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf in view of Gee.

GROUPING OF CLAIMS

(1) Appellants argue claims 1-3, 5-9, 12-14, and 17-19 as a group (App. Br. 7). We accept Appellants' grouping. We therefore treat claims 2, 3, 5-9, 12-14, and 17-19 as standing or falling with representative claim 1.

(2) Appellants argue claim 4 separately (*id.* at 11). We accept Appellants' grouping. We therefore treat claim 4 as standing or falling separately.

(3) Appellants argue claims 10, 11, and 20 as a group (*id.* at 17). We accept Appellants' grouping. We therefore treat claims 11 and 20 as standing or falling with representative claim 10.

(4) Appellants argue each of claims 15, 16 and 21 separately (*id.* at 22-27). We therefore treat these claims as standing or falling separately.

We accept Appellants' grouping of the claims. *See* 37 C.F.R. § 41.37(c)(1)(vii).

ISSUE 1

35 U.S.C. § 102(b): claims 1-3, 5-9, 12-14 and 17-19

Appellants argue their invention is not anticipated by Hilgendorf (App. Br. 7). Specifically, Appellants argue Hilgendorf does not disclose or suggest "translating instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set" as recited in claim 1 (*see also* claim 17 which has commensurate language) (*id.* at 7-11). Appellants contend Hilgendorf (i) is silent regarding a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set, as presently claimed, and (ii) converts instructions of a code A into instructions of a code B by rearranging elements of the instructions of the code A using multiplexers (*id.* at 9-10). Specifically, Appellants argue Hilgendorf is "directed to dynamic conversion between different instruction codes by a recombination of instruction elements" and thus, Hilgendorf converts and does not translate using a translation table that contains rearrangement information not a microcode memory containing sequences of instruction codes (*id.* at 7-8). Appellants further contend the translation table of Hilgendorf is not the microcode memory recited (*id.* at 9).

Furthermore, Appellants argue Hilgendorf discloses a conversion performed by using rearrangement information to rearrange the instruction elements of the initial instruction and explains that "the translation table 106 does not contain sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction

set" (*id.* at 9-10). Rather, Appellants contend, the rearrangement information controls multiplexing means which use the instruction elements of the initial code instruction as input (*id.* at 10). Appellants argue a person of ordinary skill in the field of the invention would not equate Hilgendorf's translation table containing rearrangement information for controlling multiplexers to "a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set" as recited in claim 1 (*see also* claim 17) (*id.* at 10-11).

In response, the Examiner finds that Hilgendorf's disclosure of converting microcode instructions is translating as supported by the definition of "translate" (Ans. 1, §10). The Examiner additionally finds an instruction code is an operation code and thus, finds Hilgendorf discloses a CISC instruction set which is the claimed first instruction set and the OP-codes of the CISC instruction set are the claimed instruction codes of a first instruction set (*id.*). The Examiner further finds Hilgendorf describes a microcode memory as the translation table is a memory that stores microcodes or instruction opcodes (Ans. 2-3, §10). Additionally, the Examiner finds Hilgendorf discloses the microcode memory contains sequences of instruction codes of a second instruction set that emulate the functionality of CISC instruction codes (instruction codes of the first instruction set) (*id.*).

Issue 1: Has Appellant shown that the Examiner erred in finding Hilgendorf discloses "a circuit that translates instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing

sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set?"

FINDINGS OF FACT (FF)

We find the following:

Hilgendorf Reference

(1) Hilgendorf describes an apparatus and a method for converting instructions of a code A to instruction of a code B (Abst.). Hilgendorf seeks to address the desirability of being able to process programs written in different codes (col. 1, ll. 25-27).

> (2) In a further embodiment of the invention, said instructions of code A are CISC instructions and said instructions of code B are RISC instructions.
>
> Each CISC instruction of the external instruction stream is converted to at least one internal RISC instruction. This embodiment is especially useful when a sequence of CISC instructions has to be processed by a superscalar RISC processor. As each CISC instruction usually has to be replaced by a number of RISC instructions, it is advantageous to address a table entry that contains all the necessary rearrangement information for generating the required number of RISC instructions. A fast and efficient CISC/RISC decode becomes possible. Changes of both the external CISC code or the internal RISC code can easily be implemented.

(col. 4, ll. 32-47).

(3) The process for fast instruction translation takes only one cycle and thus, instruction translation and decoding can be performed at run time (col. 3, ll. 42-44).

(4) A translation table can either be implemented as a ROM table or as a RAM table (col. 7, 11-13). Each translation table entry corresponds

to an external code A instruction; therefore, the OP-code of the external instruction can be used to determine the correct translation table entry by forwarding the OP-code to logic that converts the OP-code into the address of the corresponding entry in the table (col. 7, ll. 18-24). The address is used to access the corresponding translation table entry (col. 7, ll. 24-25). The "entry contains the OP-codes of all the internal code B instructions to which the external instruction is to be converted to, and multiplexer control information (129), which is used for controlling the recombination of the external instruction[]" (col. 7, ll. 25-30). (*See also*, Fig. 1).

> (5) The translation table (106) can be implemented as an array, with each row of said array holding the information corresponding to one code A instruction…
>
> In the example given in FIG. 3, an external instruction has to be converted to several internal instructions. The first data field of the entry holds the number n of internal code B instructions corresponding to said external code A instruction.

(col. 8, l. 61 - col. 9, l. 2).

(6) The translation table or memory contains sequences of instruction codes of the second instruction set (Fig. 3).

> (7) When converting instructions, the external instruction has to be replaced by internal instructions in a way that exactly the same operations performed by the external instructions are performed by the internal instructions, though the codes are completely different. The instructions of code B have to functionally correspond to said external instruction of code A.

(col. 5, ll. 58-64).

PRINCIPLES OF LAW

The PTO must apply the broadest reasonable meaning to the claim language, taking into account any definitions presented in the specification. *In re Cortright*, 165 F.3d 1353, 1358 (Fed. Cir. 1999). "In the absence of an express intent to impart a novel meaning to the claim terms, the words are presumed to take on the ordinary and customary meanings attributed to them by those of ordinary skill in the art." *Brookhill-Wilk 1, LLC. v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1298 (Fed. Cir. 2003) (citation omitted). "[T]he words of a claim 'are generally given their ordinary and customary meaning.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (citations omitted). The "ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application." *Id.* at 1313.

The ordinary and customary meaning of a claim term may be determined by reviewing a variety of sources, which may include the claims themselves; dictionaries and treatises. *Ferguson Beauregard v. Mega Sys., LLC*, 350 F.3d 1327, 1338 (Fed. Cir. 2003) (citations omitted). To determine the plain meaning of a claim term, one may look to extrinsic evidence so long as the extrinsic evidence does not contradict the meaning otherwise apparent from the intrinsic record. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1319 (Fed. Cir. 2005) (en banc). Accordingly, "[w]hen the intrinsic evidence is silent as to the plain meaning of a term, it is entirely appropriate … to look to dictionaries … to aid in arriving at the plain meaning of a claim term." *Helmsderfer v. Bobrick Washroom Equip., Inc.*,

527 F.3d 1379, 1382 (Fed. Cir. 2008) (construing the term "partially hidden from view").

ANALYSIS

We agree with the Examiner's definition of "translate" which we find takes the broadest reasonable meaning of the term that one of ordinary skill in the art would have attributed to it (*see* Ans. 1, §10). Thus, we find Hilgendorf discloses a circuit configured to translate instructions of a first code to instructions of a second code (FF 1). Additionally, we find the translation table is microcode memory as Hilgendorf describes the translation table as being in RAM or ROM (FF 4). Further, we agree with the Examiner (Ans. 1, §10) and find the "instruction code" recited in claim 1 is read broadly as an OP-code. Indeed, we find Appellants do not define "instruction code" in their Specification. We find none of the portions cited by Appellant provide a definition but instead provide examples or possibilities (*see, e.g.*, "may", "may be" and "[i]n one example" (Spec. 11-12)). Therefore, we find one of ordinary skill in the art at the time of the invention would reasonably find an OP-code to be an instruction code.

We also find Hilgendorf discloses a circuit that translates instruction codes of a first instruction set into addresses into a microcode memory (FF 1). We further find Hilgendorf describes a microcode memory that contains sequences of instruction codes of a second instruction set as Hilgendorf describes that an entry in the translation table contains the OP-codes of the internal code B instructions (FF 4-6). Additionally, we find Hilgendorf describes that the microcode memory contains sequences of the second

instruction set's instruction codes that emulate a functionality of the first instruction set's instruction set (FF 7).

Therefore, we are not persuaded by Appellants' arguments and find Hilgendorf discloses "a circuit configured to translate instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set" as recited in claim 1.

ISSUE 2

35 U.S.C. § 102(b): claim 4

Appellants argue Hilgendorf does not disclose that "predetermined sequences of the instruction codes of the first instruction set are used to address the microcode memory" as recited in claim 4 (App. Br. 11). Specifically, Appellants contend, the Abstract of Hilgendorf, cited by the Examiner, does not disclose a sequence of the instruction codes of the first instruction set being used to address the microcode memory (*id.* at 12). Instead, Appellants argue Hilgendorf describes that "each reference to an instruction of code A is singular, thus indicating only one instruction of code A is involved at a time" (*id.*). Appellants further contend "Hilgendorf refers to using a single instruction of code A as an input to the conversion process" (*id.* at 13). Appellants thus contend "[s]ince each translation table entry corresponds to ONE external code A instruction, it follows that…Hilgendorf does not disclose or suggest that predetermined sequences of the instruction codes of the first instruction set are used to address the microcode memory" (*id.* at 14).

The Examiner finds that Appellants are arguing timing requirements not recited in the claims (Ans. 7, §10). The Examiner additionally finds a program for a computer system has a sequence of several instructions and any sequence of instructions contains subsequences (*id.*).

Issue 2: Has the Examiner erred in finding Hilgendorf discloses "predetermined sequences of said instruction codes of said first instruction set are used to address said microcode memory?"

ANALYSIS

We find Hilgendorf discloses that the OP-code of the first instruction set is sent to logic that converts it into the address of the corresponding entry in the table (FF 4 and Fig. 1). Thus, we find the instruction codes of the first instruction set are used to address the microcode memory. Additionally, we find the predetermined sequences of the OP-code (or instruction codes) of the first instruction set is the program (FF 2). Therefore, we find Hilgendorf discloses predetermined sequences of the instruction codes of the first instruction code are used to address the microcode memory.

ISSUE 3

35 U.S.C. § 103(a): claims 10, 11, and 20

Appellants assert their invention is not obvious over Hilgendorf and Martin because Martin "appears to be silent regarding a circuit configured to detect optimizable sequences of instruction codes on-the-fly" (App. Br. 17). Moreover, Appellants contend the Examiner has provided an inadequate and "personal conclusion that '[i]t would have been obvious to one of ordinary

skill in the art at the time of the invention to be configured to detect optimizable sequences of instruction codes on-the-fly'" (*id.* at 18-19).

The Examiner finds Martin teaches that when pre-defined instruction sequences are detected, a substitute instruction is generated to replace the first instruction sequence for the desirable purpose of improving system performance (Ans. 8, §10). Additionally, the Examiner finds a skilled artisan would have been motivated to combine the techniques taught by Martin of "configuring the circuit to detect optimizable sequences of instruction codes on-the-fly" into the processor of Hilgendorf to achieve its goal of improving system performance (*id.* at 9, §10).

Issue 3: Have Appellants shown the Examiner erred in finding (a) Martin teaches a circuit configured to detect optimizable sequences of instruction codes on-the-fly and (b) one of ordinary skill in the art would have been motivated to combine the technique of Martin and Hilgendorf?

FINDINGS OF FACT

We further find as follows:

Martin Reference

> (8) Buffered, pre-fetched instructions in the instruction handling portion of a data processing system are examined to detect sequences of predetermined instructions to effect generation of a substitute instruction to be executed by an execution unit in place of the first instruction of the sequence to perform the functions specified by all of the instructions of the sequence.

(Abst.).

(9) The performance of a data processing system with instruction decode and execute overlap is improved without change to the architecture, the instruction set, the operating systems, the compilers, or the vast inventory of running production programs (col. 2, ll. 19-23). "Certain pre-defined instruction sequences are detected, and a substitute instruction is generated to replace the first instruction of the sequence, causing the execution unit of the data processing system to respond to the single substitute instruction to perform all the functions called for by the original sequence of instructions" (col. 2, ll. 23-30).

(10) Instructions are accessed and transferred to a register where they are decoded and address generation is performed (col. 2, ll. 31-33). "A sequence detector examiners the Op-code field of adjacent instructions in the instruction buffer, and various 4-bit fields of adjacent and instructions, to determine the presence of particular predefined instruction sequences" (col. 2, ll. 33-37). If a particular sequence is detected, a substitute instruction generator modifies the first instruction, substituting a "pseudo instruction" which controls "all the functions required of the execution unit to realize the results of the functions called for by the original sequence" (col. 2, ll. 37-52).

PRINCIPLES OF LAW

> [A]n implicit motivation to combine exists not only when a suggestion may be gleaned from the prior art as a whole, but when the "improvement" is technology-independent and the combination of references results in a product or process that is more desirable, for example because it is stronger, cheaper, cleaner, faster, lighter, smaller, more durable, or more efficient. … In such situations, the proper question is whether the

> ordinary artisan possesses knowledge and skills rendering him *capable* of combining the prior art references.

DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co., 464 F.3d 1356, 1368 (Fed. Cir. 2006).

Discussing the question of obviousness of a patent that claims a combination of known elements, *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007), explains:

> When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

Id. at 417. If the claimed subject matter cannot be fairly characterized as involving the simple substitution of one known element for another or the mere application of a known technique to a piece of prior art ready for the improvement, a holding of obviousness can be based on a showing that "there was an apparent reason to combine the known elements in the fashion claimed." *Id.* at 418. Such a showing requires

> some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. . . . [H]owever, the analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ.

Id. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)) (internal quotation marks omitted).

ANALYSIS

We find that Martin teaches that when certain pre-defined instruction sequences are detected, a substitution instruction is generated and replaces the instruction sequences so as to perform all the functions of the original instruction sequence (FF 9-10). We also find this change is performed "on-the-fly" as this is done during processing to eliminate some of the drawbacks of decode and execution overlap (FF 9). Thus, we find that Martin teaches "a circuit configured to detect optimizable sequences of instruction codes on-the-fly."

Additionally, we find one of ordinary skill in the art would have been motivated to combine the technique of Martin into the system of Hilgendorf. The technique of Martin is a known technique used to improve performance of a data processing system by replacing a sequence of instructions with a new instruction. We find Hilgendorf is also directed toward a computing system configured to improve processing speed and Hilgendorf also translates one instruction to another. Thus, both references are concerned with speeding up data processing and both involve changing an instruction (or a sequence of instructions) for another (or others). We therefore find one of ordinary skill in the art would have been motivated to configure the circuit to perform this technique.

ISSUE 4

35 U.S.C. § 103(a): claim 15

Appellants assert Hilgendorf does not explicitly teach the instruction codes of the first instruction set comprise Java bytecodes as recited in claim

15 (App. Br. 22). Appellants further argue (1) the Examiner "does not adequately address the issue of motivation" as Hilgendorf is silent regarding converting and executing Java code and (2) the proposed combination has not been shown to have a reasonable expectation of success (*id.* at 23). Further, Appellants contend that the Examiner did not present any support for the contention that Java bytecodes can merely be rearranged to generate host instructions (*id.*).

The Examiner finds Appellants argue the references individually (*see* Ans. 9-10, § 10). The Examiner finds that Hilgendorf describes translating instruction code of a first instruction set to instruction code of a second instruction set and one possible instruction code could be Java, as Java is well known to the computer software community (*id.* at 10, § 10). The Examiner further finds Hilgendorf is a part of a superscalar computer, superscalar computers are typically employed as web servers, and a common way of executing programs on the world wide web is by using Java bytecodes (*id.* at 10-11, § 10. Therefore, the Examiner finds one of ordinary skill in the art at the time the invention was made would have used the superscalar computer of Hilgendorf as a web server, recognized the desirability of executing Java bytecodes on a web server; and translate Java bytecodes into a host's computer codes (*id.* at 11-12, §10).

Issue 4: Have Appellants shown the Examiner erred in finding (1) Hilgendorf teaches the instruction codes of the first instruction set comprise Java bytecodes; (2) one of ordinary skill in the art would have been motivated to use Java; and (3) the proposed combination would have had a reasonable likelihood of success?

ANALYSIS

We find one of ordinary skill in the art would have recognized that Hilgendorf's system could support various programming languages. We find that using the Java language as one of the instruction sets would have been an obvious engineering decision well within the level of skilled artisans and therefore that the first instruction set's instruction codes comprise Java bytecodes. We further find converting Java (or any other computer language instructions) would have been a creative step that a person of ordinary skill in the art would have employed.

We additionally find that one of ordinary skill in the art would have been motivated to use the Java language for the instruction sets. We therefore find one of ordinary skill in the art would have been motivated to configure the circuit to perform this technique. Appellants state Java bytecodes are "an abstract, processor-independent form that cannot be directly executed by most CPUs" (Reply Br. 19) (emphasis added). Appellants themselves thus admit that some CPUs can directly execute Java bytecodes. Thus, we find that Appellants have not persuaded us that using the Java bytecodes in the system of Hilgendorf would not have had a reasonable likelihood of success.

ISSUE 5

35 U.S.C. § 103(a): claim 16

Appellants assert Gee does not teach "a circuit that comprises a portion of a Java Virtual Machine implemented in hardware" as recited in claim 16 (App. Br. 25) (emphasis omitted), but instead, a system that is

either all hardware or all software (Reply Br. 21). Moreover, according to Appellants, implementing a portion of the Java Virtual Machine in hardware provides an advantage over one entirely implemented in hardware or entirely in software (*id.*). Further, Appellants argue the Examiner has not set forth motivation to combine Hilgendorf and Gee; or presented any evidence that such a combination would have had a reasonable expectation of success (App. Br. 25-26).

The Examiner finds Gee discusses the advantages of implementing a Java Virtual Machine in hardware and thus teaches a portion of the Java Virtual Machine being implemented in hardware (Ans. 12, §10).

Issue 5: Have Appellants shown the Examiner erred in finding Gee teaches (1) the circuit comprises a portion of a Java Virtual Machine implemented in hardware; and (2) the proposed combination would have had a reasonable likelihood of success?

FINDINGS OF FACT

We further find as follows:

Gee Reference

(11) A Java run time environment, called a Java Virtual Machine (JVM), can interpret bytecodes directly or use a compiler to convert the bytecodes to a platform dependent code which may then run on the host platform directly (col. 2, ll. 42-46). The JVM has drawbacks for a real-time embedded processor application and thus, another processor may be used for generating Java bytecodes by a Java compiler and executing them directly on the platform (col. 2, ll. 48-54).

ANALYSIS

We find that Gee discloses that the Java Virtual Machine could use a compiler to convert the bytecodes and run it on a host platform directly or a separate processor for generating bytecodes and executing them on a platform (FF 11). Thus, we find Gee teaches using both hardware and software to implement a Java Virtual Machine. Accordingly, we find Gee teaches implementing a portion of a Java Virtual Machine in hardware. Since Gee teaches a Java Virtual Machine that converts bytecodes to a platform dependent code (FF 11), we find that combining the Java Virtual Machine of Gee with the system of Hilgendorf would have had a reasonable likelihood of success. Hilgendorf converts one set of instruction codes to a second set of instruction codes and Gee converts Java bytecodes (FF 1, FF 11).

ISSUE 6

35 U.S.C. § 103(a): claim 21

Appellants assert Gee does not expressly mention "a sequence of instruction codes of an instruction set comprises one or more virtual stack references" (App. Br. 27). Further, Appellants argue the Examiner has not set forth motivation to combine Hilgendorf and Gee, or presented any evidence that such a combination would have had a reasonable expectation of success (*id.*).

The Examiner finds since a Java Virtual Machine is virtual and uses stack based processing, the Java Virtual Machine must necessarily have

virtual stack references to properly execute and/or process instructions using the stack (Ans. 12-13, §10).

Issue 6: Have Appellants shown the Examiner erred in finding Gee teaches the sequences of instruction codes of the second instruction set comprise one or more virtual stack references?

ANALYSIS

We find that Gee does not teach or suggest virtual stack references. A Java Virtual Machine may use stack references; however, these stack references may or may not be virtual. We find the gap between what Gee discloses and the recited limitation is too large and would require speculation on our part. We will not engage in speculation, accordingly, we find Gee does not teach or suggest the sequences of instruction codes of the second instruction set comprise one or more virtual stack references.

CONCLUSION

Appellants have not shown the Examiner erred in finding Hilgendorf discloses "a circuit that translates instruction codes of a first instruction set on-the-fly into addresses into a microcode memory containing sequences of instruction codes of a second instruction set that emulate a functionality of the instruction codes of the first instruction set." Therefore, we affirm the Examiner's rejection of independent claim 1 as being anticipated by Hilgendorf. Since claims 2, 3, 5-9, and 12-14 depend from independent claim 1, and claim 19 depends from independent claim 18, and claims 2, 3, 5-9, 12-14, and 17-19 were not argued separately, claims 2, 3, 5-9, 12-14,

and 17-19 are found to be anticipated by Hilgendorf. Accordingly, Appellants have not shown the Examiner erred in rejecting claims 1-3, 5, 6, 8, 9, 12-14, and 17-19 under 35 U.S.C. § 102(b) as anticipated by Hilgendorf. Nor have Appellants shown the Examiner erred in rejecting claim 7 under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf.

Additionally, we find Appellants have not shown the Examiner erred in finding Hilgendorf discloses "predetermined sequences of said instruction codes of said first instruction set are used to address said microcode memory." Therefore, we affirm the Examiner's rejection of claim 4 as being anticipated by Hilgendorf.

Further, Appellants have not shown the Examiner erred in finding (a) Martin teaches a circuit configured to detect optimizable sequences of instruction codes on-the-fly, and (b) one of ordinary skill in the art would have been motivated to combine the technique of Martin and Hilgendorf. Thus Appellants have not shown the Examiner erred in rejecting claims 10, 11, and 20 under 35 U.S.C. § 103(a) for obviousness over Hilgendorf and Martin.

Further, Appellants have not shown the Examiner erred in finding (1) Hilgendorf teaches the instruction codes of the first instruction set comprise Java bytecodes; (2) one of ordinary skill in the art would have been motivated to use Java; and (3) the proposed combination would have a reasonable likelihood of success. Accordingly, Appellants have not shown the Examiner erred in rejecting claim 15 under 35 U.S.C. § 103(a) for obviousness over Hilgendorf.

Additionally, Appellants have not shown the Examiner erred in concluding Gee teaches (1) the circuit comprises a portion of a Java Virtual

Machine implemented in hardware; and (2) the proposed combination would have a reasonable likelihood of success. Thus we find that Appellants have not shown the Examiner erred in rejecting claim 16 under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf in view of Gee.

Finally, however, Appellants have shown the Examiner erred in concluding Gee teaches the sequences of instruction codes of the second instruction set comprise one or more virtual stack references. Thus we find that Appellants have shown the Examiner erred in rejecting claim 21 under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf in view of Gee.

DECISION

The Examiner's rejection of claims 1-6, 8, 9, 12-14, and 17-19 under 35 U.S.C. § 102(b) as being anticipated by Hilgendorf is affirmed.

The Examiner's rejection of claims 10, 11, and 20 under 35 U.S.C. § 103(a) as being obvious over Hilgendorf and Martin is affirmed.

The Examiner's rejection of claims 7 and 15 under 35 U.S.C. § 103(a) as being obvious over Hilgendorf is affirmed.

The Examiner's rejection of claim 16 and under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf in view of Gee is affirmed.

The Examiner's rejection of claim 21 under 35 U.S.C. § 103(a) as being unpatentable over Hilgendorf in view of Gee is reversed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv) (2009).

AFFIRMED-IN-PART

nhl

CHRISTOPHER P MAIORANA, PC
LSI Corporation
24840 HARPER
SUITE 100
ST CLAIR SHORES MI 48080